

IN THE SPECIFICATION:

Please replace the following paragraphs.

On page 6, beginning with line 3:

Java Embedded Server Module (i.e., a plug-in module) (110) for the Forte for Java<sup>™</sup> IDE (90), as shown in Figure 6, contains tools which facilitates creation of Java Embedded Server bundles. The ~~plug-in module~~ Java Embedded Server Module (110) contains a code template tool (112), a Java Embedded Server Manifest Generator tool (114) to help in creating a valid Java Embedded Server manifest file for the bundle, and a Jar Packager tool (116) which helps packaging up the bundle. Also, links to web pages (118) containing documentation and other information about Java Embedded Server are included. Further, access to Java Embedded Server Swing-based tools from within the module may be included.

An embodiment of the plug-in Java Embedded Server module (110) is accessed via a drop-down menu in Forte for Java<sup>™</sup> (90) entitled 'Java Embedded Server' (150) as shown in Figure 8. As can be seen, the drop-down menu consists of three submenus entitled 'Manifest Generator' (152), 'Jar Packager' (154), and 'Product Info and Downloads' (156) which provide access to a selected feature of the Java Embedded Server module (110) identified by submenu name. The submenus are displayed by clicking on the 'Java Embedded Server' menu name (150) with the mouse (79). By continuing to hold the mouse button down and dragging the mouse pointer over the desired submenu name, the submenu name is selected upon release of the mouse button. The result of selecting the submenu is the launching of the application to perform the selected submenu feature by the system.

Referring to Figure 7, an example for the usage of ~~plug-in module~~ Java Embedded Server module (110) by a bundle-writer for developing Java Embedded Server bundles follows.

On page 7, beginning with line 1:

In order to accomplish this, a valid manifest file for the Java Embedded Server bundle must be written. The bundle-writer uses the Java Embedded Server Manifest Generator tool (114) to create a valid manifest file (ST134). Next, the bundle-writer uses the Jar Packager tool (116) to create the bundle's jar file (ST136). The bundle-writer must then debug the bundle. Using the Forte for Java<sup>TM</sup> debugger (98), the bundle-writer can correct any problems that need to be fixed (ST138). Using the supplied links within the 'Java Embedded Server' menu (150) to 'Product Info and Downloads' (156) which provides online Java Embedded Server documentation (156), the bundle-writer can find the material that may be a resource to what changes need to be made (ST142).

On page 8, beginning with line 5:

An embodiment of the Java Embedded Server Manifest Generator tool (114), as shown in Figure 10, is an aid in creating valid manifest files for Java Embedded Server bundles. The Manifest Generator (114) saves development time by providing ~~the name~~ of each Java Embedded Server manifest header name (175) 183 along with the description (170), the syntax (172), and an example of the use of the particular manifest header (174) (e.g., Bundle-Name). To start the Java Embedded Server Manifest Generator (114), the 'Manifest Generator' menu item (152) is selected from the 'Java Embedded Server' menu (150). Two text boxes are located near the top of the screen which hold the descriptions (170) and syntax (172) of the manifest header (174), a listing

of each Java Embedded Server manifest header name ~~(175)~~ 183 to the left and its corresponding manifest header value (176) in the text field (177) ~~for the Manifest Header Value (176)~~ to the right. Also, action buttons (181) appear at the bottom of the screen.

When first starting the Java Embedded Server Manifest Generator (114), text is displayed inside each of the two help boxes at the top of the screen. These help boxes display the description (170) and syntax (172) of each manifest header (183) ~~(174)~~. To view the help file on a particular manifest header name ~~(175)~~ 174, the name of the particular manifest header ~~(175)~~ 174 is selected for the left help box or the corresponding text field (177) is selected from the right help box. The selection is made by clicking or tabbing to the area. A brief description (170) of the manifest header is displayed in the box labeled 'Manifest Header Description' (170) and the syntax (172) is displayed in the box labeled 'Manifest Header Syntax.'

A particular header name ~~(175)~~ (174) in the generated manifest file is included when the header's checkbox (178) is selected. When checking a header name (183) ~~(174)~~, the corresponding text field background changes from gray to white. The value of the header ~~(174)~~ (176) is entered into the text field (177) immediately to the right of the header name (183) ~~(175)~~. Another way to include a header (183) ~~(174)~~ into the manifest file is to type the value (176) into the text field (177). As soon as the manifest header value (176) is entered into the text field (177), a checkbox (178) gets checked.